

REMARKS/ARGUMENTS

The foregoing amendment and the following arguments are provided to impart precision to the claims, by more particularly pointing out the invention, rather than to avoid prior art.

35 U.S.C. § 112 Rejections

Examiner rejected claims 1 and 19 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Appropriate correction has been made with the foregoing amendment.

35 U.S.C. § 103(a) Rejections

Examiner rejected claims 1-2, 4-7, 10-11, 13-16, 19-20, and 22-24 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 5,301,325 (hereinafter “Benson”) in view of U.S. Patent 5,668,999 (hereinafter “Gosling”).

Applicant’s independent claims 1, 10, and 19 include limitations not disclosed in Benson in view of Gosling. Therefore, applicant’s independent claims are patentable over Benson in view of Gosling.

In particular, as amended, independent claims 1, 10, and 19 include the claim limitation, or a limitation similar thereto, of performing only one test for the block of code to determine if an architectural stack is available, the block of code including multiple instructions adding data to the stack or removing data from the stack. (emphasis added.) (Applicant’s claim 1.)

Neither Benson nor Gosling discloses the claimed limitation. Rather, Benson is limited to disclosing “tracking” stack usage within routines.

By tracking stack usage within routines, the compiler can distinguish up-level stack and return address references from valid local references.

For each routine being compiled, the compiler builds a flow graph and visits each basic block in flow order, beginning at the routine entry point. The compiler records the amount of which the stack pointer is changed in each block, and maintains the cumulative offset from the routine entry point. As it processes each instruction in the block, it can use this cumulative offset, along with any stack-based operand specifiers in the instruction (or stack reference implicit in the instruction) . . .

In each of these cases, the compiler / translator detects these occurrences so that user can be advised of the specific usage, and thus the user can make the appropriate changes to the source code. (emphasis added.) (Benson, col. 4, lines 31-61.)

Therefore, Benson is not testing at a start of said block of code to determine if resources of an architectural stack that are needed by a block of code, as is claimed by applicant. Rather, Benson is processing each instruction individually and maintaining a cumulative offset of the stack to inform the user during a binary translation so that the user can make the appropriate changes to the source code.

Therefore, applicant’s claims include limitations not disclosed nor suggested in Benson or Gosling. As a result, applicant’s claims are patentable over Benson in view of Gosling.

Furthermore, the remaining claims depend from at least one of the independent claims discussed above, and therefore include the distinguishing limitations of the independent claims. As a result, applicant's remaining claims are also patentable over Benson in view of Gosling.

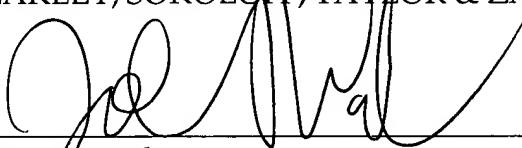
CONCLUSION

Applicants respectfully submit the present application is in condition for allowance. If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call John Ward at (408) 720-8300, x237.

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN


John P. Ward
Reg. No. 40,216

Date: November 16, 2004

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300